

## N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM  
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT  
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED  
IN THE INTEREST OF MAKING AVAILABLE AS MUCH  
INFORMATION AS POSSIBLE

"Made available under NASA sponsorship  
in the interest of early and wide dis-  
semination of Earth Resources Survey  
Program information and without liability  
for any use made thereof."

~~T77-1367~~  
JSC-12709  
8.0-10263  
NAS CR-  
16669

**"AS-BUILT" DESIGN SPECIFICATION  
FOR  
PHASE III MODEL AREAS ADDED TO  
THE MONTHLY DATA BASE OF THE U.S.**

Job Order 74-963

(E80-10263) AS-BUILT DESIGN SPECIFICATION  
FOR PHASE 3 MODEL AREAS ADDED TO THE MONTHLY  
DATA BASE OF THE US (Lockheed Electronics  
Co.) 28 p HC A03/MF A01

N80-30846

CSCL 05B

Unclass

G3/43 00263

Prepared By  
Lockheed Electronics Company, Inc.  
Systems & Services Division  
Houston, Texas  
Contract NAS 9-15200  
For  
EARTH OBSERVATIONS DIVISION  
SCIENCE AND APPLICATIONS DIRECTORATE



*National Aeronautics and Space Administration*  
**LYNDON B. JOHNSON SPACE CENTER**

*Houston, Texas*

March 1977

LEC-10354

JSC-12709

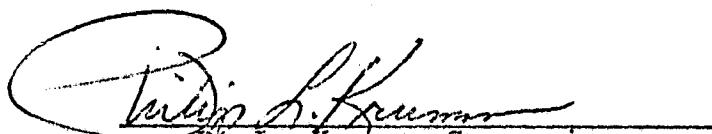
"AS-BUILT" DESIGN SPECIFICATION  
FOR  
PHASE III MODEL AREAS ADDED TO  
THE MONTHLY DATA BASE OF THE U.S.

Job Order 74-963

PREPARED BY

K. Williams

APPROVED BY



P. L. Krumm, Supervisor  
Applications Software Section

Prepared By  
Lockheed Electronics Company, Inc.  
For  
Earth Observations Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

March 1977

LEC-10354

## CONTENTS

Section	Page
1. SCOPE . . . . .	1-1
2. APPLICABLE DOCUMENTS . . . . .	2-1
3. SYSTEM DESCRIPTION . . . . .	3-1
3.1 <u>HARDWARE DESCRIPTION</u> . . . . .	3-1
3.2 <u>DATA BASE STRUCTURE</u> . . . . .	3-1
3.2.1 DATA BASE STORAGE REQUIREMENTS . . . . .	3-1
3.2.2 CONTROL AND DIRECTORY BLOCKS . . . . .	3-1
3.2.3 DATA DESCRIPTORS AND DATA BLOCKS . . . . .	3-1
3.2.4 MODEL DEFINITION BLOCK . . . . .	3-1
4. OPERATION . . . . .	4-1
4.1 <u>DATA BASE INITIALIZATION AND DEFINITION</u> . . . . .	4-1
4.2 <u>DATA CONVERSION</u> . . . . .	4-1
4.3 <u>DATA BASE LOADING</u> . . . . .	4-1
4.4 <u>DATA BASE LISTING</u> . . . . .	4-1
APPENDIX	
A. <u>STRUCTURE</u> . . . . .	A-1
B. <u>VARIABLE CODES</u> . . . . .	B-1
C. <u>CONVERSION PROGRAM FLOWCHARTS AND LISTINGS</u> . . . . .	C-1

## 1. SCOPE

This document updates the existing monthly weather and yield data bases for the United States by inserting data for additional model areas in Montana, South Dakota, Oklahoma, Nebraska and Colorado. The data base design is identical to that previously documented for the U.S. in the "As-Built Design Specification for the Yield Estimation Subsystem (YES) Monthly Yield Data Base and Supporting Programs" (JSC-12537/LEC-10034).

## **2. APPLICABLE DOCUMENTS**

- o Action Documentation 63-1347-4963-12**
- o "As-Built" Design Specification for the Yield Estimation Subsystem (YES) Monthly Data Base and Supporting Programs" (JSC-12537/LEC-10034).**

### 3. SYSTEM DESCRIPTION

#### 3.1 HARDWARE DESCRIPTION

These data and supporting programs previously documented in "As-Built Design Specification" (JSC-12537/LEC-10034), are resident on the IBM 360/195 complex at Suitland, Maryland. They should be transferable to any IBM 360-370 series machine with sufficient disk to handle the data base and main memory to support the PL/I optimizing compiler.

#### 3.2 DATA BASE STRUCTURE

The monthly weather and yield data base is a tree structure, with nodes, or levels, being the country, region, zone and strata. The four respectively represent: the United States, the Great Plains, states and crop reporting districts (CRD's).

##### 3.2.1 DATA BASE STORAGE REQUIREMENTS

The data base storage requirements remain the same as those for the U.S. described in "As-Built Design Specification."

##### 3.2.2 CONTROL AND DIRECTORY BLOCKS

The format for the control and directory blocks follows that previously defined in "As-Built Design Specification." The U.S. directory block originally contained the additional CRD's, and no modification to either control or directory block was necessary.

##### 3.2.3 DATA DESCRIPTORS AND DATA BLOCKS

The format for the data descriptors and data blocks follows that previously defined in "As-Built Design Specification."

##### 3.2.4 MODEL DEFINITION BLOCK

Structure is provided to allow inclusion of model definitions.

## 4. OPERATION

### 4.1 DATA BASE INITIALIZATION AND DEFINITION

Data base initialization and definition were accomplished using the supporting programs documented in "As-Built Design Specification" (JSC-12537/LEC-10034). Only data descriptor definition was required, as the original U.S. control block and directory definitions had reserved space for these additional CRD's.

### 4.2 DATA CONVERSION

Data for all new United States CRD's located at CCEA in Columbia, Missouri, exist in English units and in a format different from that required for inclusion in the data base. Eight programs were written to handle the necessary conversions to the required format:

- METOME (Montana, South Dakota meteorological data),
- METTWO (Oklahoma, Nebraska meteorological data),
- METCOLO (Colorado meteorological data),
- YLDONE (Oklahoma, Nebraska yield data),
- YLDTWO (South Dakota yield data),
- COLOYLD (Colorado yield data),
- MONTYLD (Montana yield data),
- DEGDAZE (degree days for all states).

Converted data were stored on a catalogued disk file, and transferred to the data base disk file when all conversions were completed.

Sample listings and flowcharts appear in Appendix C.

### 4.3 DATA BASE LOADING

Data for new model areas were loaded using the updating program UPDDATA, documented in "As-Built Design Specification."

### 4.4 DATA BASE LISTING

Three programs documented in "As-Built Design Specification" accomplish listing of control block and directory information (YESLS02 and YESLS04, respectively), as well as the data itself (LISTJOB).

**APPENDIX A**  
**STRUCTURE**

## United States Data Year Entry

There is a maximum of 47 years following the data descriptor entry in a data block for each United States region; each year entry is 128 bytes long.

DCL 1 US,

2 YEAR	FIXED BIN(15,0),
2 NXTYRREC	FIXED BIN(15,0),
2 NXTYRDISP	FIXED BIN(15,0),
2 FILLER	FIXED BIN(15,0),
2 MEANTEMP(12)	FIXED BIN(15,0),
2 PRECIP(12)	FIXED BIN(15,0),
2 DEGREEDAY(12)	FIXED BIN(15,0),
2 HARVESTED(4)	FIXED BIN(31,0),
2 PLANTED(4)	FIXED BIN(31,0),
2 PRODUCTION(4)	FIXED BIN(31,0);

**APPENDIX B**  
**VARIABLE CODES**

### Meteorological Variables

Precipitation	5
Mean temperature	35
Degree days above	40

### Yield Variables

Harvested	101
Planted	102
Production	103

### Crops

Spring wheat	201
Winter wheat	202

### Units of Measurement

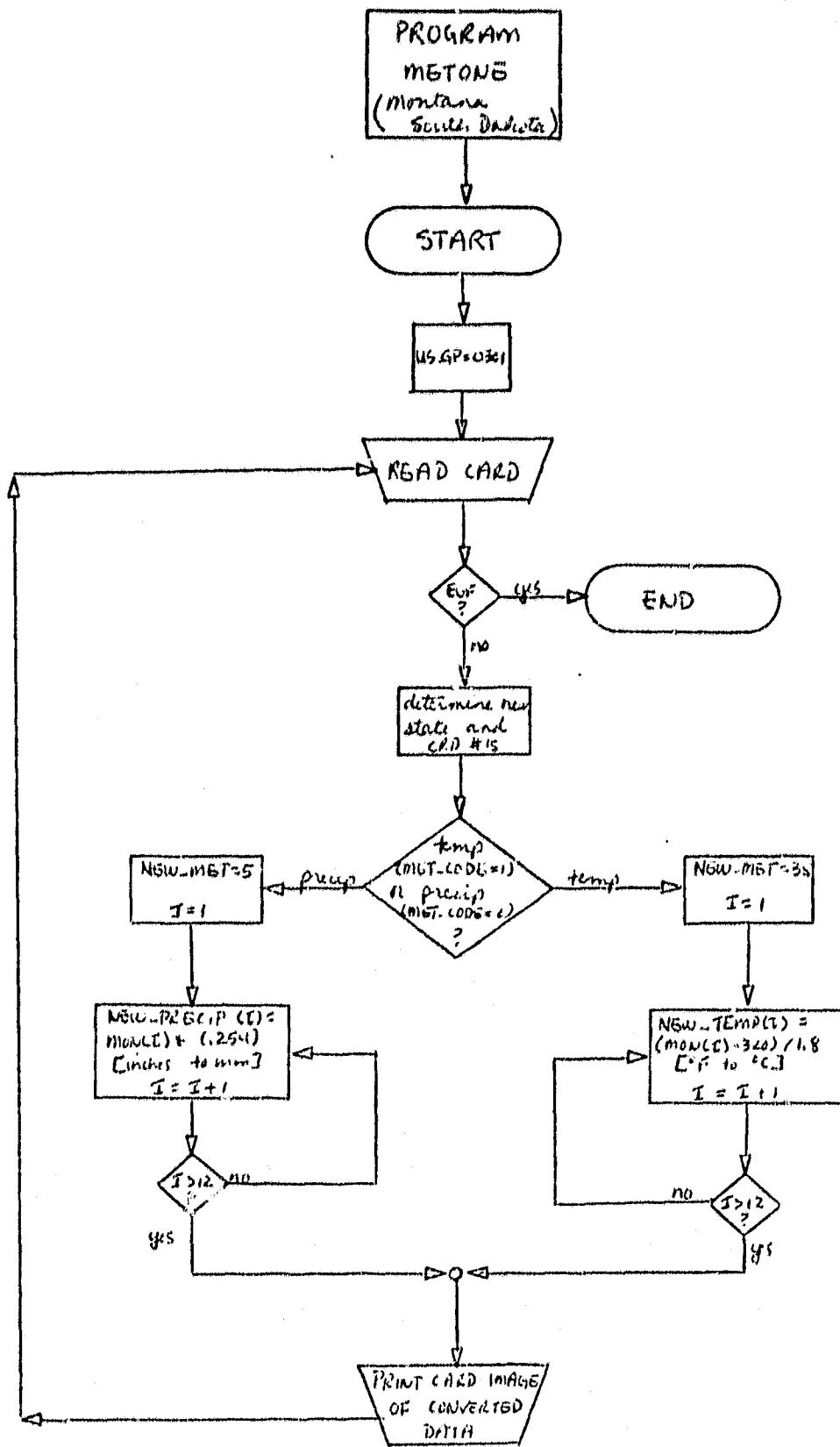
Monthly	5
Millimeters	201
Quintals	228
Hectares	236
Degrees Centigrade	241

### Others

Monthly	26
Year	61
Pointer	90
Record pointer	91
Displacement pointer	92
Filler	99

B-1 8

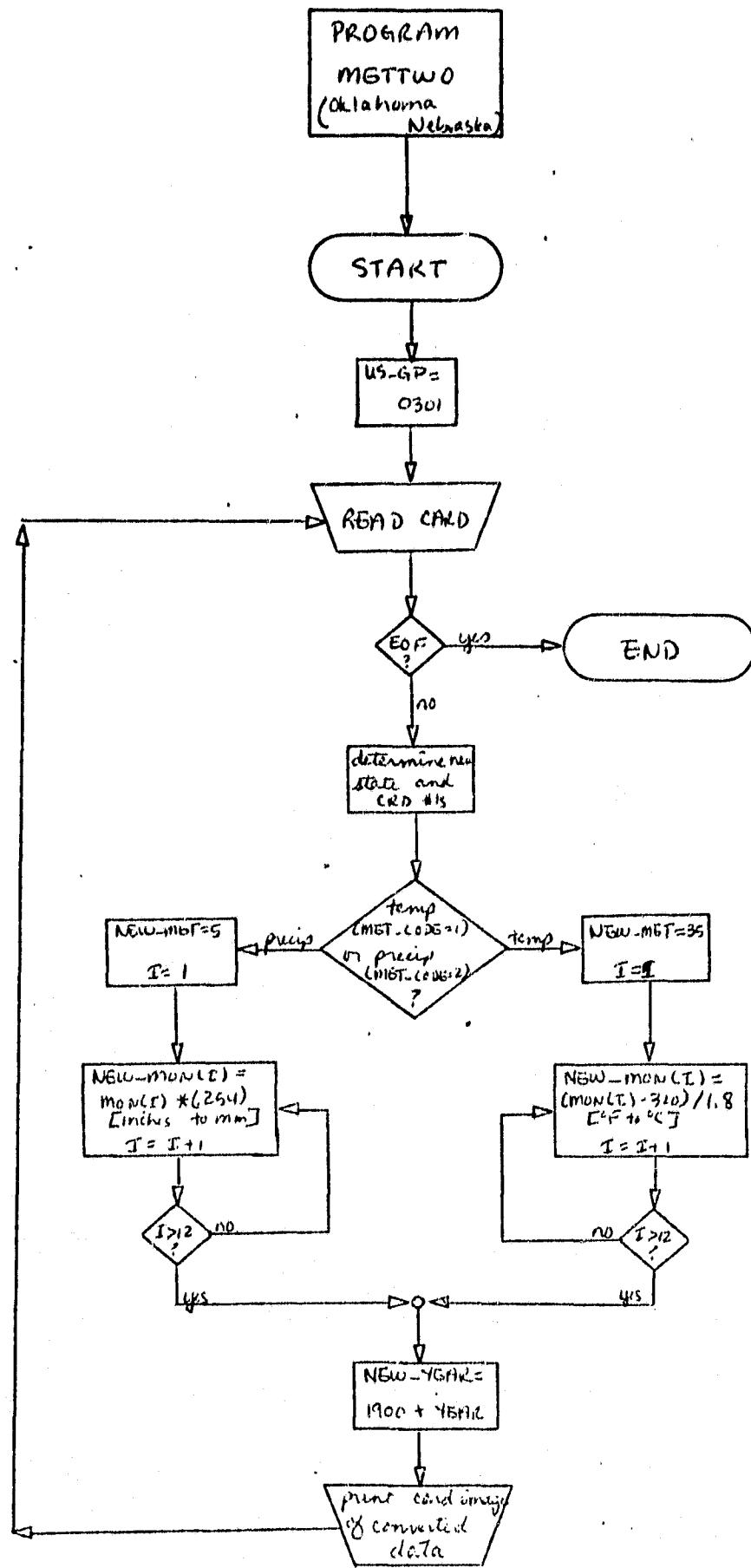
**APPENDIX C**  
**CONVERSION PROGRAM FLOWCHARTS**  
**AND LISTINGS**



```

METONE: PROC OPTIONS (MAIN); YEAR,MON(12) FIXED(5);
DCL (MET_CODE,STATE,CRD#) FIXED(5);
DCL (NEW_YEAR,NEW_MONTH,NEW_PRECIP(12)) FIXED(7,2);
DCL (NEW_STATE,CRD#) FIXED(6);
DCL SYSFILE STREAM OUTPUT;
US_GP = 0301;
READ: GET SKIP EDIT (MET_CODE,STATE,CRD#,YEAR,MON(I),DO_I=1 TO 12);
      (F(1,0),2 F(2,0)F(4,0),12 F(5,0));
      IF STATE = 99 THEN GO TO EOF;
      IF STATE = 25 AND CRD# = 1 THEN NEW_STATE_CRD = 301000;
      IF STATE = 24 AND CRD# = 1 THEN NEW_STATE_CRD = 301000;
      IF STATE = 25 AND CRD# = 2 THEN NEW_STATE_CRD = 307000;
      IF STATE = 24 AND CRD# = 2 THEN NEW_STATE_CRD = 307000;
      IF STATE = 40 AND CRD# = 4 THEN NEW_STATE_CRD = 469000;
      IF STATE = 39 AND CRD# = 4 THEN DO;
      IF MET_CODE = 1 THEN DO;
      DO I = 1 TO 12;
      NEW_TEMP(I) = (MON(I)-320)/1.8;
      END;
      PUT EDIT (YEAR,NEW_MET,NEW_TEMP(I),DO_I=1 TO 12);
      US_GP,NEW_STATE_CRD,(X(2),FT4,0),F(3,0),12 F(5,0);
      F(4,0),F(5,0);
END;
ELSE IF MET_CODE = 2 THEN DO;
      DO J = 1 TO 12;
      NEW_PRECIP(J) = (MON(J) * (.254));
      END;
      PUT EDIT (YEAR,NEW_MET,NEW_PRECIP(J),DO_J=1 TO 12);
      US_GP,NEW_STATE_CRD,(X(2),FT4,0),F(3,0),12 F(5,0);
      F(4,0),F(5,0);
END;
GO TO READ;
EOF: END METONE;

```

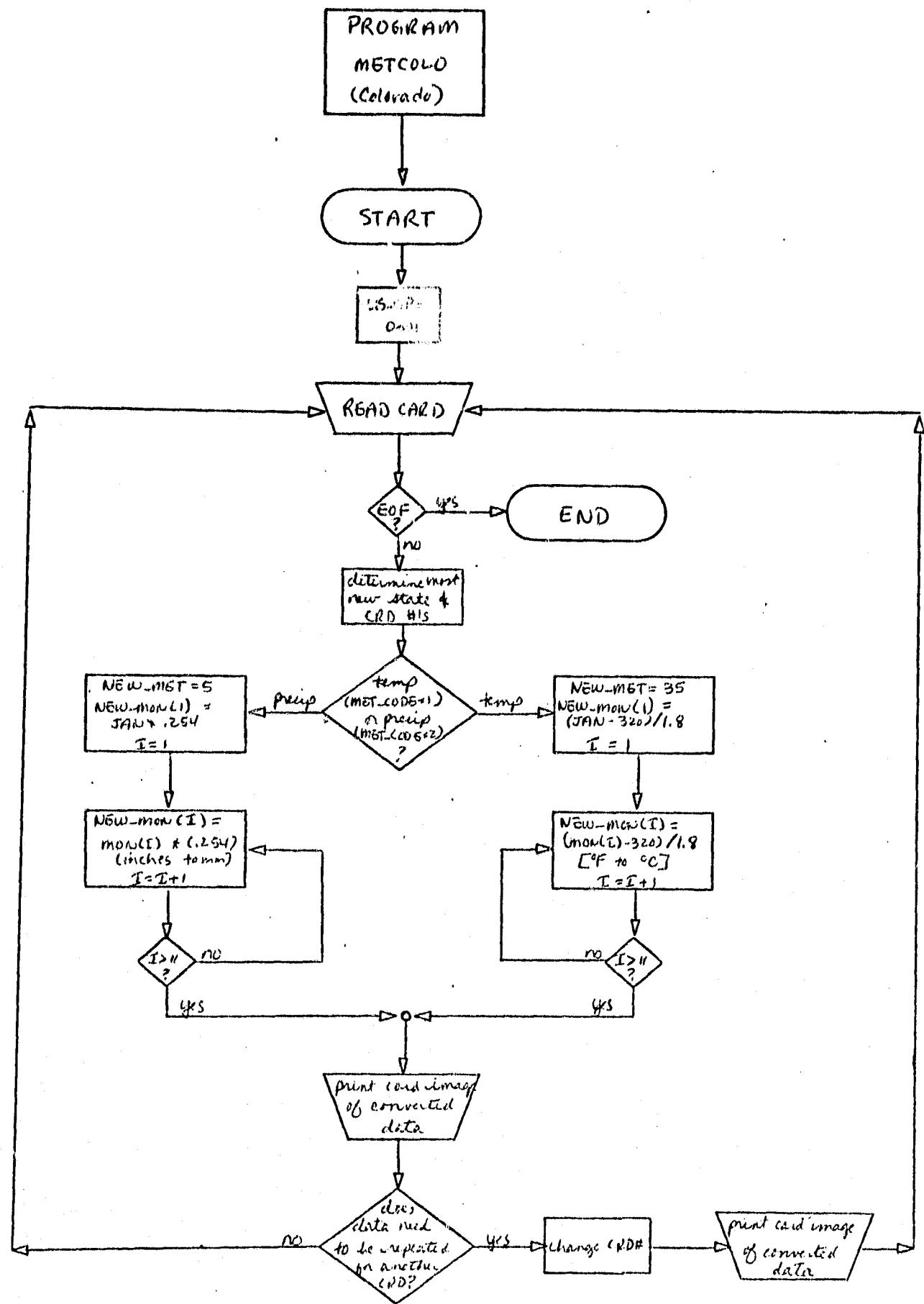


C-3  
12

```

METTWO: PROC OPTIONS(MAIN);
DCL (MFT_CODE•STATE•CRD#•YEAR•MON(12)) FIXED(5);
DCL (NEW_MET•NEW_YEAR•US_GP•J) FIXED(5);
DCL NEW_MON(12) FIXED(7•2);
DCL (NEW_STATE_CRD•STATE_CRD•NEW) FIXED(6);
DCL SYSPPINT FILE STREAM-OUT3;
US_GP = 0301;
J = 0;
READ: GET SKIP EDIT(MET_CODE•STATE•CRD#•YEAR•(MON(I)-DO_I=1-TO-12));
(F(1•0),3,F(2•0),X(1),12,F(5•0));
IF STATE = 99 THEN DO TO FOR;
IF STATE = 34 & CRD# = 3 THEN NEW_STATE_CRD = 407000;
IF STATE = 34 & CRD# = 6 THEN NEW_STATE_CRD = 408000;
IF STATE = 34 & CRD# = 9 THEN NEW_STATE_CRD = 409000;
IF STATE = 25 & CRD# = 2 THEN NEW_STATE_CRD = 312000;
IF STATE = 25 & CRD# = 3 THEN NEW_STATE_CRD = 313000;
IF NEW_YEAR = 1900 + YEAR;
IF MET_CODE = 1 THEN DO;
NEW_MET = 3;
DO_I = 1 TO 12;
DO NEW_MON(I) = (MON(I)-320)/1.5;
END;
PUT EDIT(NEW_YEAR•NEW_MET•(NEW_MON(I)) DO_I=1 TO 12),
US_GP•NEW_STATE_CRD)(XT2),F(4•0),F(3•0),12,F(5•0),
F(4•0),F(6•0));
END;
ELSE IF MET_CODE = 2 THEN DO;
NEW_MET = 5;
DO_I = 1 TO 12;
DO NEW_MON(I) = (MON(I) * (.254));
END;
PUT EDIT(NEW_YEAR•NEW_MET•(NEW_MON(I)) DO_I=1 TO 12),
US_GP•NEW_STATE_CRD)(XT2),F(4•0),F(3•0),12,F(5•0),
F(4•0),F(6•0));
END;
GO TO READ;
EOF: END METTWO;
/* EOF: END METTWO;
//GO•SYSPRINT DD USN=4•EDS•CCEA•STPLAINS•DISP=MOD
//GO•SYSIN DD *;
```

ORIGINAL PAGE IS  
OF POOR QUALITY



## PL/I OPTIMIZING COMPILER

METCOL0: PROC OPTIONS(MAIN);

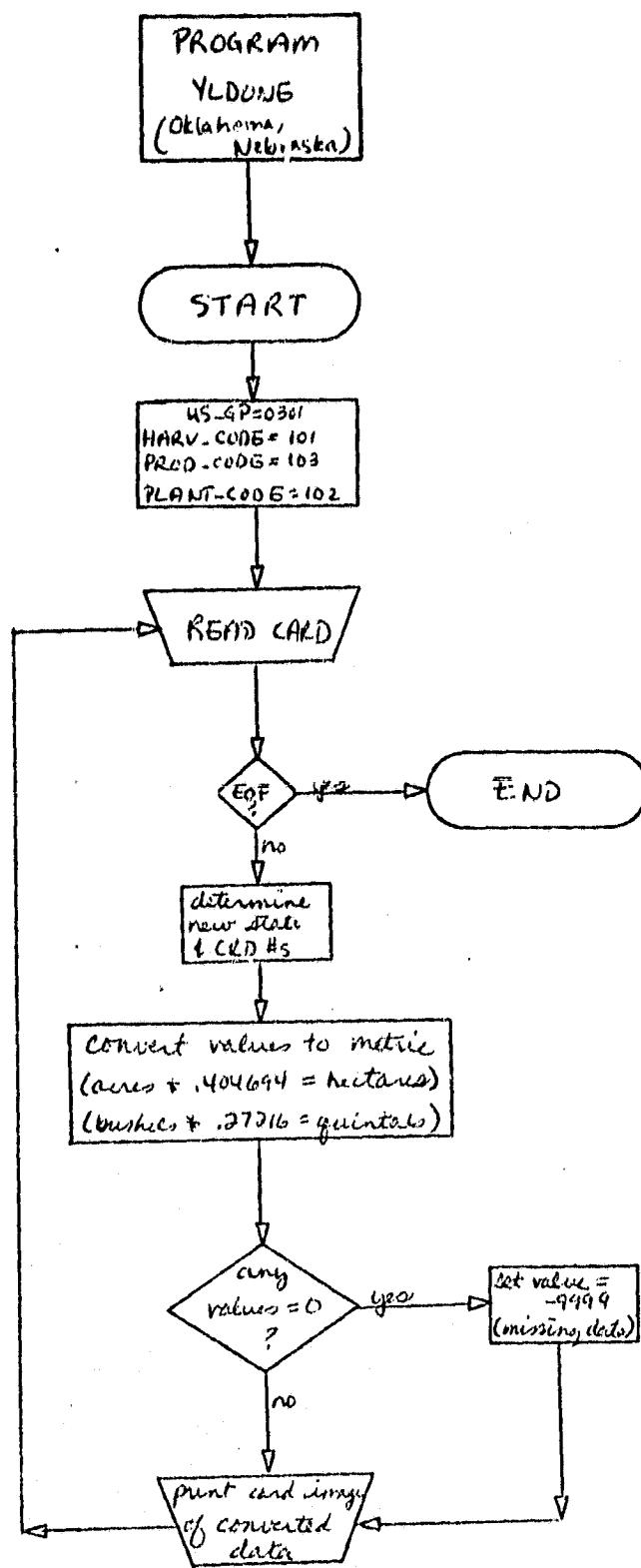
## SOURCE-LISTING

STRT

```

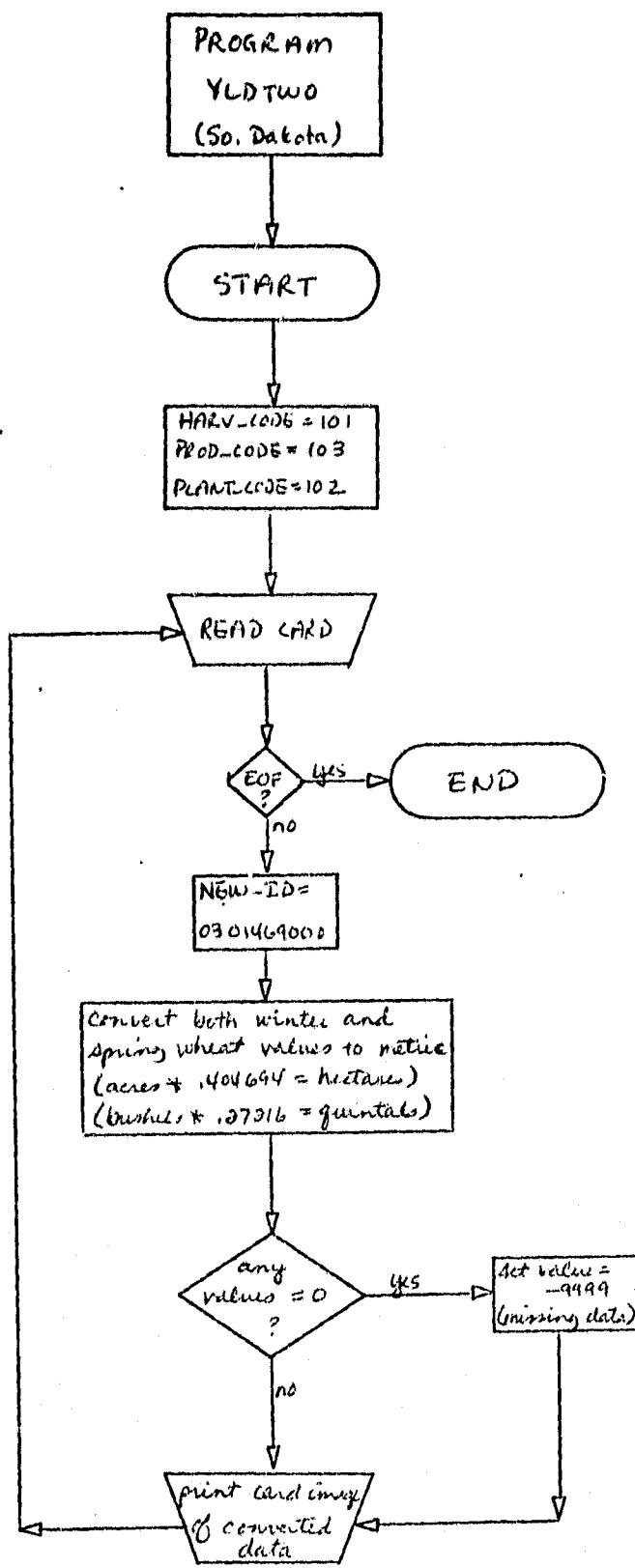
1  METCOL0: PROC OPTIONS(MAIN);
2    DCL (MET_CODE,STATE,CDF#,YEAR,JAN,MON(11),NEW_MET) FIXED(5);
3    DCL NEW_MON(12) FIXED(7,2);
4    DCL (NEW_ID,ALT_ID) FIXED(10);
5    DCL SYSPRINT FILE STREAM OUTPUT;
6    READ: GET SKIP EDIT(MET_CODE,STATE,CDF#,YEAR,JAN,
7      (MON(1) DO I=1 TO 11) ) F(1,0),2-F(2,0),2-F(4,0),11-F(5,0);
8    IF STATE = 99 THEN GO TO EOF;
9    IF CDF# = 2 THEN NEW_ID = 0301081000;
10   IF CDF# = 5 THEN NEW_ID = 0301085000;
11   IF MET_CODE = 1 THEN DO;
12     NEW_MET = 35;
13     NEW_MON(1) = (JAN-320)/1.8;
14     DO I = 1 TO 11;
15       NEW_MON(I+1) = (MON(I)-320)/1.8;
16     END;
17     PUT EDIT(YEAR,NEW_MET,1) NEW_MON(1) DO I=1 TO 12;
18   END;
19   ELSE IF MET_CODE = 2 THEN DO;
20     NEW_MET = 5;
21     NEW_MON(1) = JAN * (.254);
22     DO I = 1 TO 11;
23       NEW_MON(I+1) = (MON(I)) * (.254);
24     END;
25     PUT EDIT(YEAR,NEW_MET,1) NEW_MON(1) DO I=1 TO 12;
26   END;
27   IF NEW_ID = 0301081000 THEN DO;
28     ALT_ID = 0301087000;
29     PUT EDIT(YEAR,NEW_MET,1) NEW_MON(1) DO I=1 TO 12;
30     ALT_ID) (X(2),F(4,0),F(3,0)*T2 F(5,0)*F(10,0));
31   END;
32   GO TO READ;
33   EOF: END METCOL0;

```



YLDONE: PROC OPTIONS (MAIN); FIXED(5);  
 DCL (STATE,CRD#\*YEAR, CODE, PROD\_CODE, PLANT\_CODE) FIXED(5);  
 DCL (USGP,CHARV,PLANT) FIXED(10);  
 DCL (CHARV,PROD,PLANT) FIXED(6);  
 DCL NEW STATE CRD FIXED(6);  
 DCL (NHARV,NPROD,NPLANT) FIXED(12,2);  
 DCL SYSPRINT FILE STREAM OUTPUT;  
 DS GP = 0301;  
 HARV CODE = 101; PROD\_CODE = 103; PLANT\_CODE = 102;  
 READ: GET SKIP (2,F(2,0)) X(2) \* F(4,0) \* 3 F(10,0);  
 IF STATE = 25 THEN GO TO 101;  
 IF STATE = 25 CRD# = 2 THEN NEW STATE CRD = 312000;  
 IF STATE = 25 CRD# = 3 THEN NEW STATE CRD = 313000;  
 IF STATE = 25 CRD# = 4 THEN NEW STATE CRD = 315000;  
 IF STATE = 25 CRD# = 5 THEN NEW STATE CRD = 316000;  
 IF STATE = 25 CRD# = 6 THEN NEW STATE CRD = 317000;  
 IF STATE = 25 CRD# = 7 THEN NEW STATE CRD = 318000;  
 IF STATE = 25 CRD# = 8 THEN NEW STATE CRD = 319000;  
 IF STATE = 25 CRD# = 9 THEN NEW STATE CRD = 406000;  
 IF STATE = 34 CRD# = 7 THEN NEW STATE CRD = 407000;  
 IF STATE = 34 CRD# = 8 THEN NEW STATE CRD = 408000;  
 NHARV = HARV \* (404694);  
 NPROD = PROD \* (27216);  
 NPLANT = PLANT \* (-404694);  
 IF NPROD = 0 THEN NP40D = -9999;  
 IF NHARV = 0 THEN NHARV = -9999;  
 IF NPLANT = 0 THEN NPLANT = -9999;  
 PUT EDIT (YEAR, HARV CODE, NHARV, USGP, NEW STATE CRD)  
 (X(2)\*F(4,0)\*F(3,0)\*F(10,0)\*X(50)\*F(4,0)\*F(2,0));  
 PUT EDIT (YEAR, PROD CODE, NPROD, USGP, NEW STATE CRD)  
 (X(2)\*F(4,0)\*F(3,0)\*F(10,0)\*X(50)\*F(4,0)\*F(6,0));  
 PUT EDIT (YEAR, PLANT CODE, NPLANT, USGP, NEW STATE CRD)  
 (X(2)\*F(4,0)\*F(3,0)\*F(10,0)\*X(50)\*F(4,0)\*F(7,0));  
 GO TO READ;  
 EOF: END YLDONE;

ORIGINAL PAGE IS  
OF POOR QUALITY



Q-9

YLDT 40: PROC OPTIONS(MAIN) FIXED(5);  
 DCL (STATE,CRD#,PROD\_CODE,PLANT\_CODE) FIXED(5);  
 DCL NEW\_ID FIXED(10);  
 DCL (SHARY,SPROD,SPLANT,WHARY,WPROD,WPLANT) FIXED(110) INIT(0);  
 DCL (NEW\_SPROD,NEW\_SPLANT) FIXED(12,2) INIT(0,0);  
 DCL (NEW\_WHARY,NEW\_WPROD,NEW\_WPLANT) FIXED(12,2) INIT(0,0);  
 DCL SYSPRINT FILE STREAM OUTPUT;  
 HARY CODE = 101; PROD\_CODE = 103; PLANT\_CODE = 102;  
 READ: GET SKIP EDIT (STATE,CRD#,YEAR,SHARY,SPROD,SPLANT,WHARY,WPROD,  
 WPLANT) (2F(2,0),X(2),F(4,0),6,(10,0));

---

IF STATE = 99 THEN 60 TO EOF;

---

IF STATE = 39 & CRD# = 3 THEN NEW\_ID = 0301469000;

---

NEW\_SHARY = SHARY \* (.404694) \* (.27216);

---

NEW\_SPROD = SPROD \* (.404694);

---

NEW\_WHARY = WHARY \* (.404694);

---

NEW\_WPROD = WPROD \* (.27216);

---

NEW\_WPLANT = WPLANT \* (.404694);

---

IF NEW\_SHARY = 0 THEN NEW\_SHARY = -9999;

---

IF NEW\_SPROD = 0 THEN NEW\_SPROD = -9999;

---

IF NEW\_SPLANT = 0 THEN NEW\_SPLANT = -9999;

---

IF NEW\_WHARY = 0 THEN NEW\_WHARY = -9999;

---

IF NEW\_WPROD = 0 THEN NEW\_WPROD = -9999;

---

IF NEW\_WPLANT = 0 THEN NEW\_WPLANT = -9999;

---

PUT EDIT (YEAR,SHARY,CODE,NEW\_SPLANT,WHARY,WPROD,WPLANT) (X(2),F(4,0),F(3,0),T2F(10,0),X(40),F(10,0),F(10,0),F(10,0));

---

PUT EDIT (YEAR,SPROD,CODE,NEW\_SPROD,NEW\_WPROD,NEW\_WPLANT) (X(2),F(4,0),F(3,0),T2F(10,0),X(40),F(10,0),F(10,0));

---

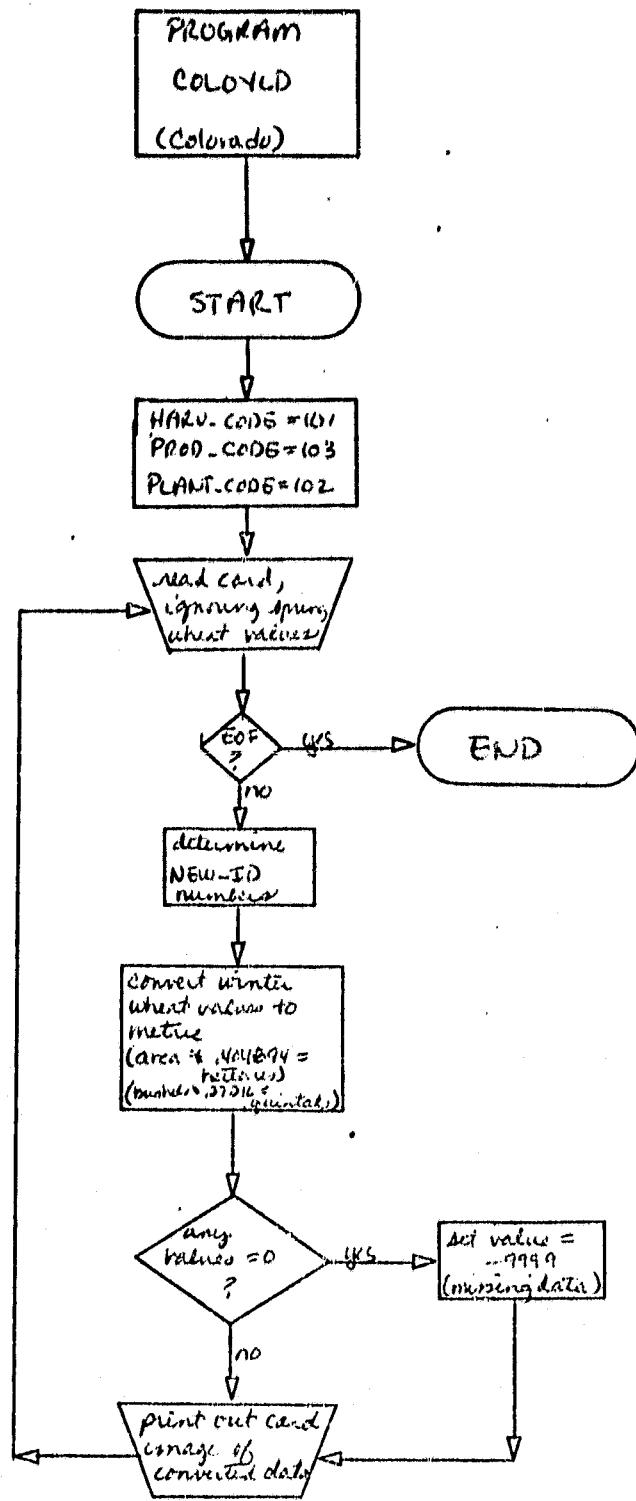
PUT EDIT (YEAR,PLANT\_CODE,NEW\_SPLANT,NEW\_WPLANT,NEW\_ID) (X(2),F(4,0),F(3,0),T2F(10,0),X(40),F(10,0));

---

GO TO READ;

---

EOF: END YLDT NO;

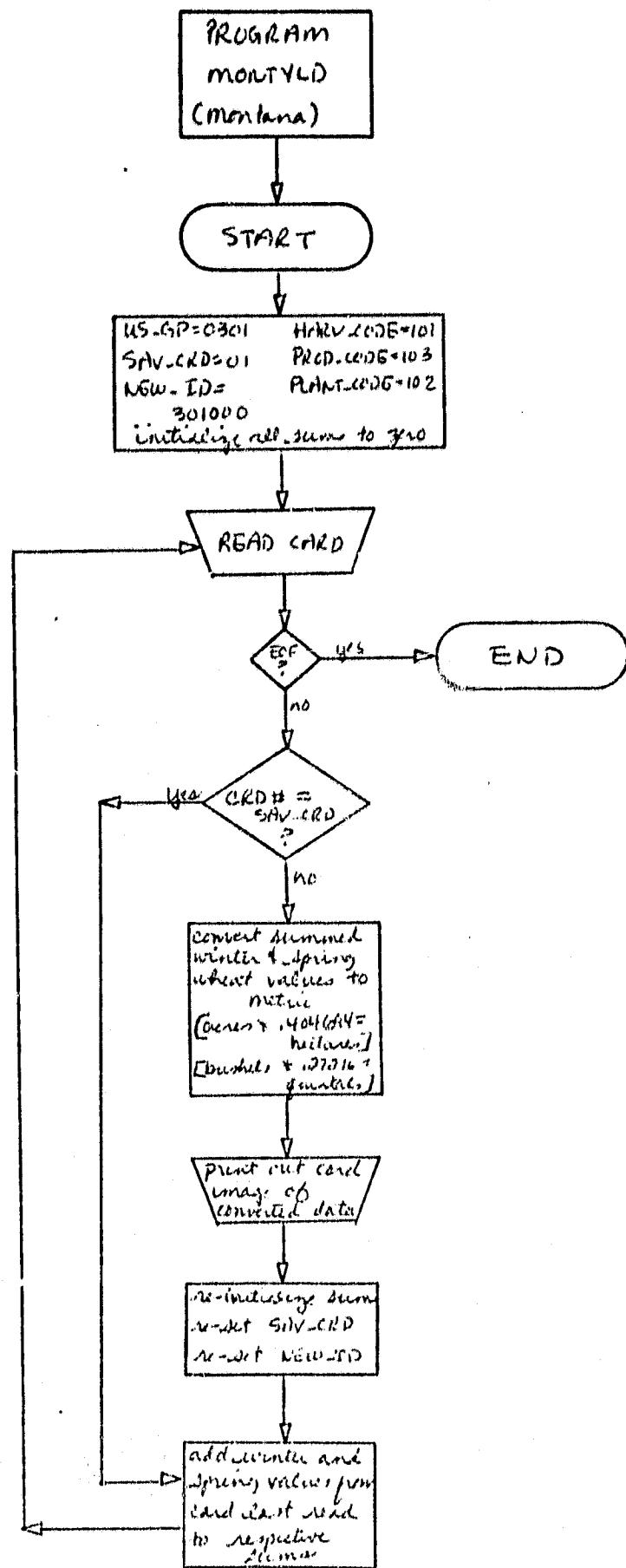


```

COLOYLD: PROC OPTIONS(MAIN):
DCL (STATE*CRD#*YEAR) FIXED(5);
DCL (HARV CODE *WPROD *WPLANT) FIXED(10);
DCL (NEW-ID *WHARV *NEW-WPLANT) FIXED(12,2) INIT(0.0);
DCL (SYSPRINT FILE STREAM OUTPUT) FIXED(12,2) INIT(0.0);

HARV CODE = 101: PROD CODE = 103: PLANT CODE = 102;
REAN: GET SKIP EDIT STATE(CRD#*YEAR*WHARV*WPROD*WPLANT);
      ((2 F(2,0)*X(2)*F(4,0)*X(30)*3 F(10,0));
      IF STATE = 99 THEN GO TO EOF;
      IF STATE = 5 & CRD# = 1 THEN NEW-ID = 0301081000;
      IF STATE = 5 & CRD# = 7 THEN NEW-ID = 0301087000;
      IF STATE = 5 & CRD# = 4 THEN NEW-ID = 0301088000;
      IF CRD# = 2 THEN NEW-ID = 0301082000;
      IF CRD# = 9 THEN NEW-ID = 0301086000;
      IF CRD# = 7 & WHARV # 74694: THEN NEW-ID = 0301089000;
      NEW-WPLANT = WPLANT * (.404694);
      NEW-WPLANT = WPLANT * (.27216);
      IF NEW-WHARV = 0 THEN NEW-WHARV = -9999;
      IF NEW-WPROD = 0 THEN NEW-WPROD = -9999;
      IF NEW-WPLANT = 0 THEN NEW-WPLANT = -9999;
      PUT EDIT (YEAR*HARV*CODE*NEW-WPLANT*NEW-ID)
            ((X(2)*F(4,0)*F(3,0)*F(11,0)*X(50)*F(10,0));
            PUT EDIT (YEAR*PROD*CODE*NEW-WPROD*NEW-ID)
            ((X(2)*F(4,0)*F(3,0)*F(10,0)*X(50)*F(10,0));
            PUT EDIT (YEAR*PLANT*CODE*NEW-WPLANT*NEW-ID)
            ((X(2)*F(4,0)*F(3,0)*F(10,0)*X(750)*F(10,0));
      GO TO READ;
EOF: END COLOYLD;

```



## PL/I OPTIMIZING COMPILER

MONTHLY: PROC OPTIONS(MAIN);

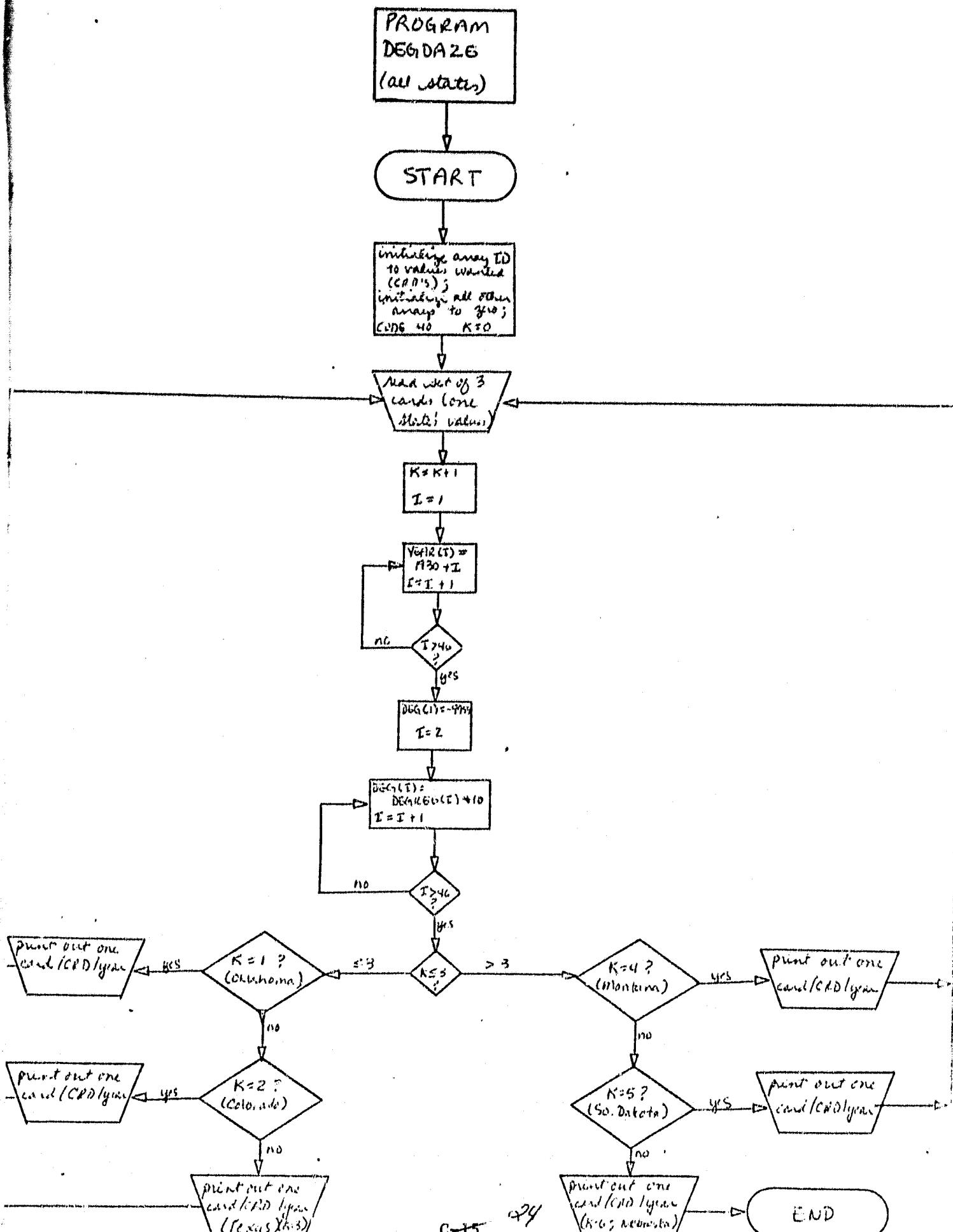
## SOURCE LISTING

### STAT

```

1  MONTYLD: PROC OPTIONS(MAIN);
2    DCL (STATE•CODE•COUNTY•YEAR•US_GPD•FILED•SAV_CRD)•FIXED(5);
3    DCL (SHARV•CODE•CRD•NEW_ID)•FIXED(6);
4    DCL (NEW_STATE•CRD•SPLANT)•FIXED(10) INIT(0);
5    DCL (SUM_SHARV•SU4_SPROD•SUM_SPLANT)•FIXED(10) INIT(0);
6    DCL (SUM_SHARV•SU4_SPROD•SUM_SPLANT)•FIXED(10) INIT(0);
7    DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
8    DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
9    DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
10   DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
11   DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
12   DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
13   DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
14   DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
15   DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
16   DCL (NEW_SHARV•NF4_SPROD•NEW_SPLANT)•FIXED(12,2) INIT(0);
17   READ: GET SKIP EDIT(STATE•CRD#•COUNTY#•YEAR•SHARV•SPROD•SPLANT,
18     IF STATE = 99 THEN GO TO EOF;
19     IF CRD# = SAV_CRD THEN DO;
20       NEW_SHARV = SUM_SHARV * (.404694);
21       NEW_SPROD = SUM_SPROD * (.27216);
22       NEW_SPLANT = SUM_SPLANT * (.404694);
23       NEW_SHARV = SUM_SHARV * (.404694);
24       NEW_SPROD = SUM_SPROD * (.27216);
25       NEW_SPLANT = SUM_SPLANT * (.404694);
26       PUT EDIT(YEAR•SHARV•CODE•NEW_SHARV•US_GPD•NEW_ID)
27         ((X(2)*F(4,0))•F(3,0)•Z(F(10,0)*X(40))•F(6,0));
28       PUT EDIT(YEAR•SPROD•CODE•NEW_SPROD•US_GPD•NEW_ID)
29         ((X(2)*F(4,0))•F(3,0)•Z(F(10,0)*X(40))•F(6,0));
30       PUT EDIT(YEAR•SPLANT•CODE•NEW_SPLANT•US_GPD•NEW_ID)
31         ((X(2)*F(4,0))•F(3,0)•Z(F(10,0)*X(40))•F(6,0));
32       IF CRD# = 01 THEN NEW_STATE_CRD = 301000;
33       IF CRD# = 02 THEN NEW_STATE_CRD = 302000;
34       IF CRD# = 03 THEN NEW_STATE_CRD = 303000;
35       IF CRD# = 04 THEN NEW_STATE_CRD = 304000;
36       IF CRD# = 05 THEN NEW_STATE_CRD = 305000;
37       IF CRD# = 06 THEN NEW_STATE_CRD = 306000;
38       IF CRD# = 07 THEN NEW_STATE_CRD = 307000;
39       IF CRD# = 08 THEN NEW_STATE_CRD = 308000;
40       IF CRD# = 09 THEN NEW_STATE_CRD = 309000;
41       NEW_ID = NEW_STATE_CRD;
42       END;
43       SUM_SHARV = SUM_SHARV + SHARV;
44       SUM_SPROD = SUM_SPROD + SPROD;
45       SUM_SPLANT = SUM_SPLANT + SPLANT;
46       SUM_SHARV = SUM_SHARV + SHARV;
47       SUM_SPROD = SU4_SPROD + SPROD;
48       SUM_SPLANT = SUM_SPLANT + SPLANT;

```



ORIGINAL PAGE IS  
OF POOR QUALITY

```
DGAZE: PROC OPTIONS(MAIN);
DCL (CODE,K) FIXED(5);
DCL I0(13) FIXED(10) INIT(3014060009*301407000*301408000*301409000,
      301081000*301087000*301088000*301450000*301301000*301307000,
      301459000*301312000*301313000);
DCL DEGREF(46) FIXED(5,1) INIT(46) V.0;
DCL DEG(46) FIXED(5) INIT(46) V.0;
DCL YEAR(46) FIXED(5) INIT(46) V.0;
DCL SYSPRINT FILE STREAM OUTPUT;
CODE = 40; K = 0;
ON ENDFILE GO TO EOJ;
READ: GET SKIP EDIT(DEGREE(I), D0-I, =1 TO -46)
      (2(15 F(5,1), SKIP)*14 F(5,1));
K = K + 1;
DO I = 1 TO 46; = 1930 + I;
END;
DEG(1) = -9999;
DO I = 2 TO 46;
  DEG(I) = DEGREE(I) -* 10;
END;
IF K < 4 THEN DO;
  IF K = 1 THEN DO J = 1 TO 4;
    DO I = 1 TO 46;
      PUT EDIT(YEAR(I), CODE, DEG(I), ID(J))
        (X(2)*F(4,0)*F(3,0)*X(20), F(5,0)*X(35), F(10,0));
    END;
    END;
    ELSE IF K = 2 THEN DO J = 5 TO 7;
      DO I = 1 TO 46;
        PUT EDIT(YEAR(I), CODE, DEG(I), ID(J))
          (X(2)*F(4,0)*F(3,0)*X(20), F(5,0)*X(35), F(10,0));
      END;
    END;
    ELSE IF K = 3 THEN DO I = 1 TO 46;
      IF K = 4 THEN DO J = 9 TO 10;
        DO I = 1 TO 46;
          PUT EDIT(YEAR(I), CODE, DEG(I), ID(J))
            (X(2)*F(4,0)*F(3,0)*X(20), F(5,0)*X(35), F(10,0));
        END;
      END;
      ELSE IF K = 5 THEN DO I = 1 TO 46;
        IF K = 4 THEN DO J = 9 TO 10;
          DO I = 1 TO 46;
            PUT EDIT(YEAR(I), CODE, DEG(I), ID(J))
              (X(2)*F(4,0)*F(3,0)*X(25), F(5,0)*X(30), F(10,0));
          END;
        END;
        ELSE IF K = 6 THEN DO J = 12 TO 13;
          DO I = 1 TO 46;
            PUT EDIT(YEAR(I), CODE, DEG(I), ID(J))
              (X(2)*F(4,0)*F(3,0)*X(25), F(5,0)*X(30), F(10,0));
          END;
        END;
      END;
    END;
EOJ: END READE;
```